

Configuration Manual

MSc Research Project
MSc Data Analytics

Mohammad Farooque Azam
Student ID: X21198926

School of Computing
National College of Ireland

Supervisor:
Dr. Anh Duong Trinh (Senja)

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Mohammad Farooque Azam.....
Student ID: x21198926.....
Programme: MSc Data Analytics..... **Year:** 2022-23.....
Module: MSc Research Project.....
Lecturer: Dr. Anh Duong Trinh (Senja).....
Submission Due Date: 14/08/2023.....
Project Title: Multi scale context aware drug review sentiment analysis using pretrained MedBERT
Word Count: 676..... **Page Count:** 10.....

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Mohammad Farooque Azam.....
Date: 14/08/2023.....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Mohammad Farooque Azam
Student ID: X21198926

1 Introduction

The goal of the document is to give stepwise direction to replicate the work of “Multi scale context aware drug review sentiment analysis using pretrained med-BERT.de”.

2 Hardware requirements

The project was implemented under below mentioned hardware requirements:

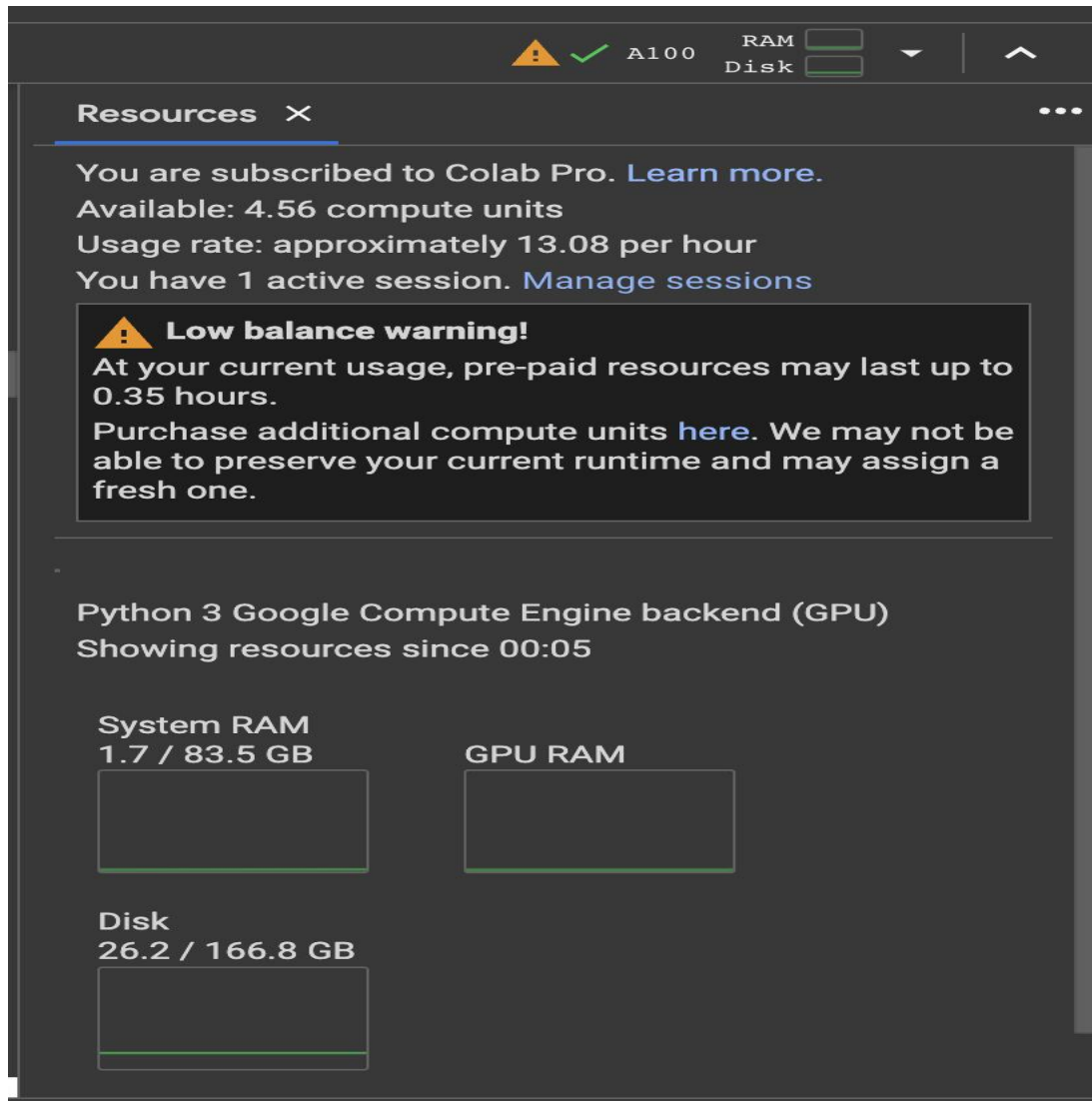
2.1 Local Machine configuration



Figure 1: Local machine hardware configuration.

2.2 GPU Configuration

The local machine hardware specification were not sufficient for the execution of project modules. GPU and High ram access were purchased from Google Collab and most of the part of project was executed on this high hardware specification.



The screenshot displays the 'Resources' tab in Google Colab Pro. At the top, there are status indicators: a warning icon, a green checkmark, 'A100', and progress bars for 'RAM' and 'Disk'. The main content area shows subscription information: 'You are subscribed to Colab Pro. Learn more.', 'Available: 4.56 compute units', 'Usage rate: approximately 13.08 per hour', and 'You have 1 active session. Manage sessions'. A prominent warning box states: 'Low balance warning! At your current usage, pre-paid resources may last up to 0.35 hours. Purchase additional compute units here. We may not be able to preserve your current runtime and may assign a fresh one.' Below this, the hardware configuration is listed as 'Python 3 Google Compute Engine backend (GPU)' with 'Showing resources since 00:05'. Three progress bars are shown: 'System RAM' at 1.7 / 83.5 GB, 'GPU RAM' (unlabeled with a value), and 'Disk' at 26.2 / 166.8 GB.

Resources X

You are subscribed to Colab Pro. [Learn more.](#)
Available: 4.56 compute units
Usage rate: approximately 13.08 per hour
You have 1 active session. [Manage sessions](#)

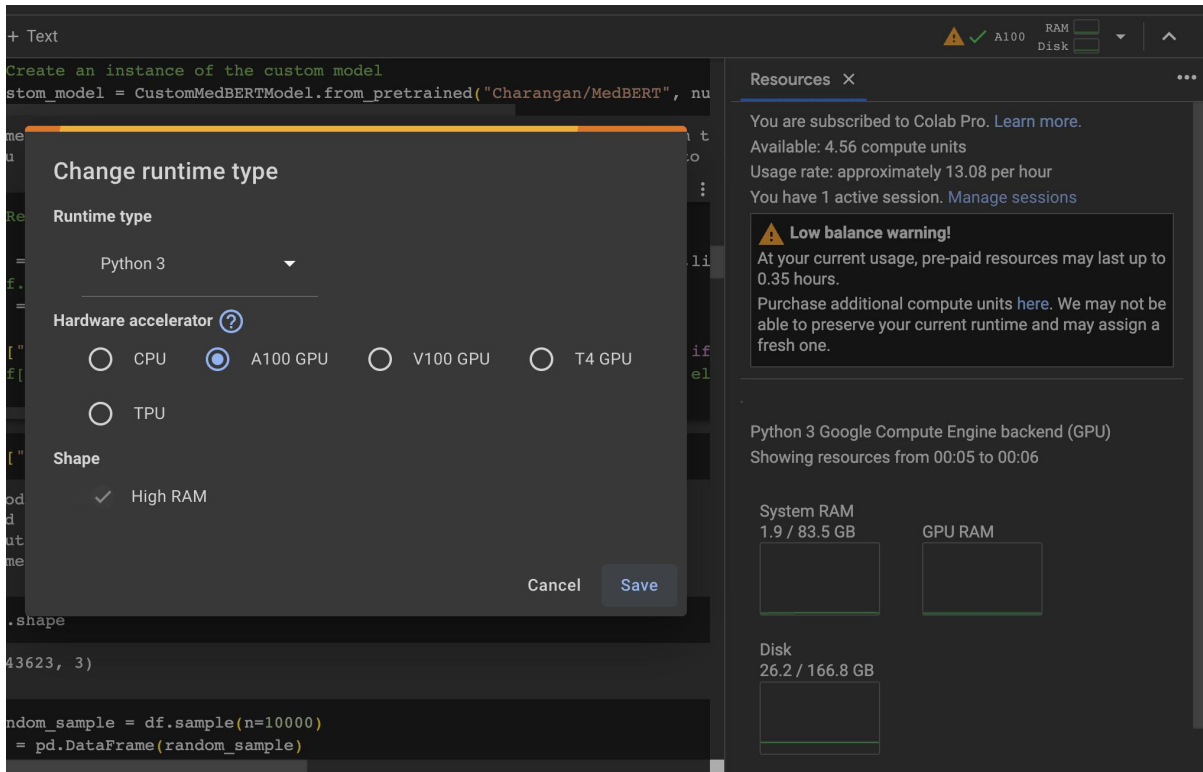
⚠ Low balance warning!
At your current usage, pre-paid resources may last up to 0.35 hours.
Purchase additional compute units [here](#). We may not be able to preserve your current runtime and may assign a fresh one.

Python 3 Google Compute Engine backend (GPU)
Showing resources since 00:05

System RAM
1.7 / 83.5 GB

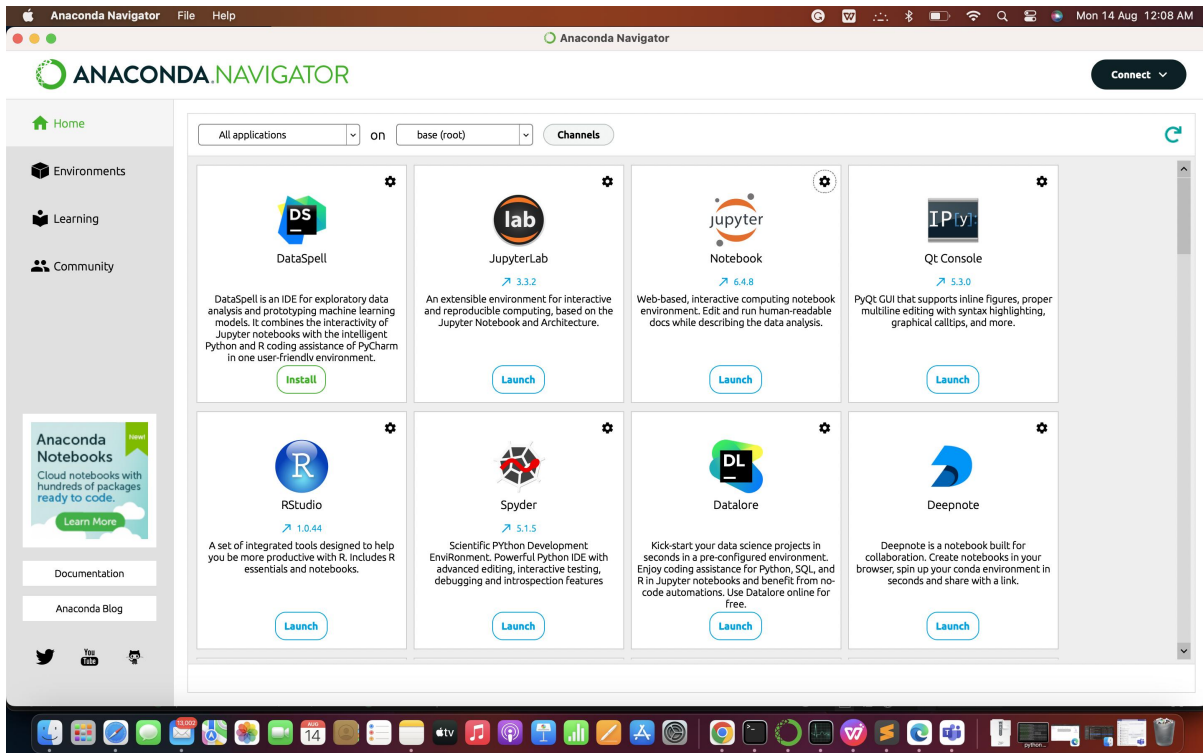
GPU RAM

Disk
26.2 / 166.8 GB



3 Software requirements

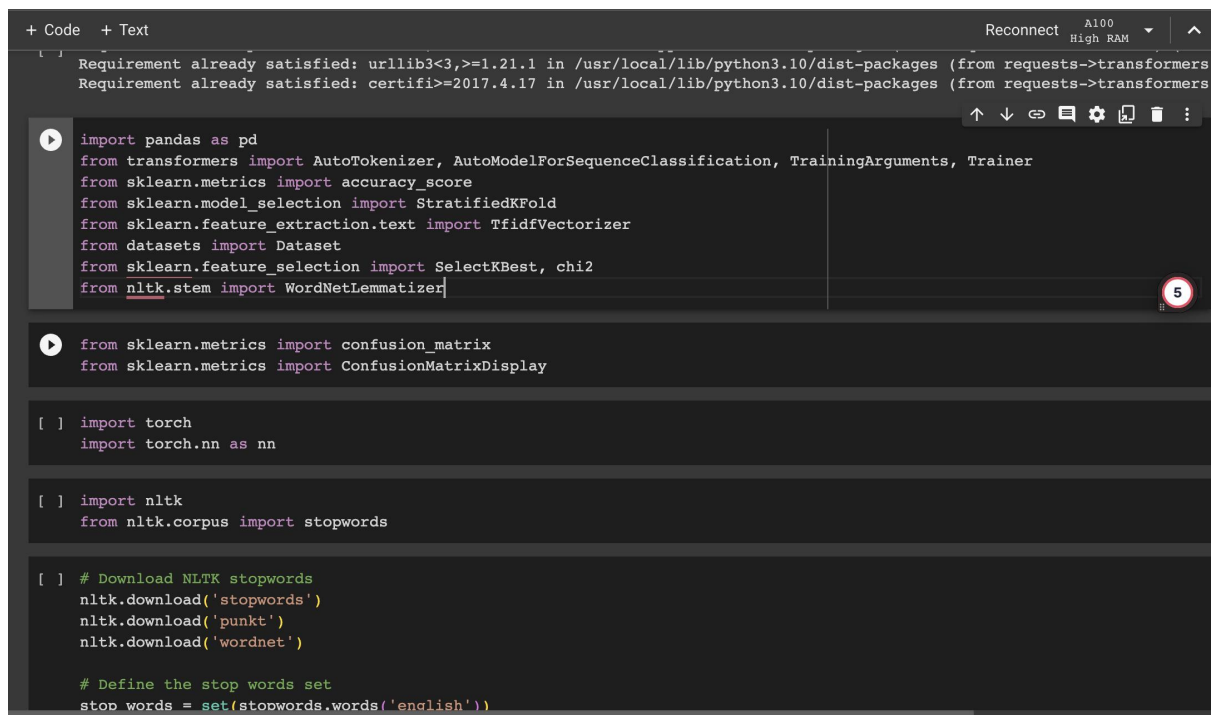
Anaconda Navigator is required to load Jupyter Notebook and execute Python codes on local machine.



3.1 Software requirements

Package Requirements The python packages were installed using pip and conda command in jupyter notebook and Google collab environment respectively. The packages installed are listed below:

- pandas
- numpy
- torch
- pathlib
- matplotlib
- pip install datasets evaluate transformers
- pip install transformers[torch]
- pip install accelerate -U
- pip install nltk transformers



```
+ Code + Text Reconnect A100 High RAM ^
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->transformers)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->transformers)

import pandas as pd
from transformers import AutoTokenizer, AutoModelForSequenceClassification, TrainingArguments, Trainer
from sklearn.metrics import accuracy_score
from sklearn.model_selection import StratifiedKFold
from sklearn.feature_extraction.text import TfidfVectorizer
from datasets import Dataset
from sklearn.feature_selection import SelectKBest, chi2
from nltk.stem import WordNetLemmatizer

from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay

[ ] import torch
import torch.nn as nn

[ ] import nltk
from nltk.corpus import stopwords

[ ] # Download NLTK stopwords
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')

# Define the stop words set
stop_words = set(stopwords.words('english'))
```

4 Dataset Description

Out of three datasets to be used, two of them are obtained from UCI Machine Learning Repository. UCI Machine Learning is a public data source licensed to be used for academic purposes. The third dataset WebMed is taken from kaggle, which is also a public data source for research works. Drugs.com has over 0.25 million rows and have DrugName, Condition, Review, Rating, Date and Useful count as main columns. The DrugLib.com training dataset has 3107 comments and the test dataset has 1036 posts, each with 9 features. The datasets are publicly available on below mentioned link for analysis:

Data source 1:-

<https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Drugs.com%29>

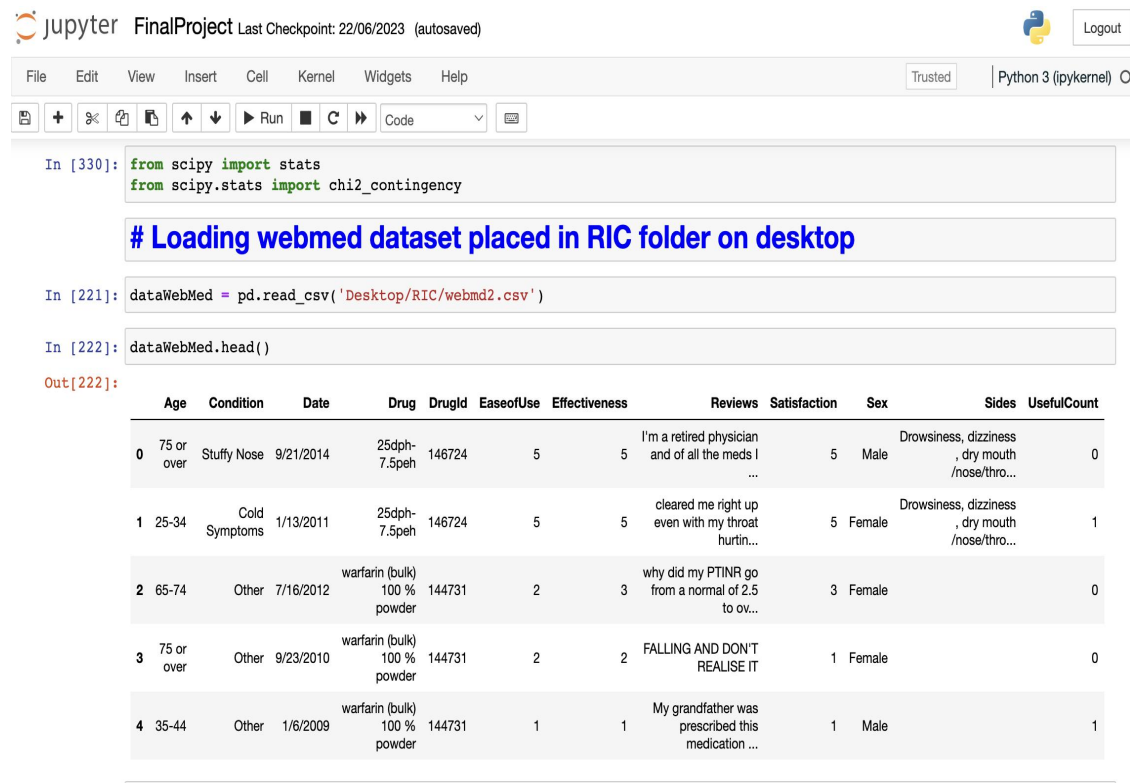
Data source 2:- <https://www.kaggle.com/datasets/rohanharode07/webmd-drug-reviews-dataset>

5 Model Preparation

The FinalProject_Data_cleaning.ipynb file available in the artefacts zip file contains the code implementation to install libraries, loading models and predicting the classification of drug reviews. The code also contains implementation to save the trained model and results in a separate file.

5.1 data cleaning and preprocessing steps

- The datasets are loaded.



The screenshot shows a Jupyter Notebook interface with the following content:

```
In [330]: from scipy import stats
from scipy.stats import chi2_contingency
```

Loading webmed dataset placed in RIC folder on desktop

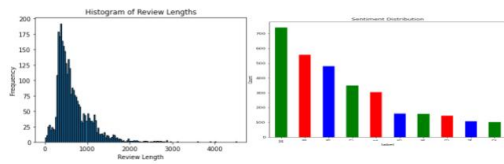
```
In [221]: dataWebMed = pd.read_csv('Desktop/RIC/webmd2.csv')
```

```
In [222]: dataWebMed.head()
```

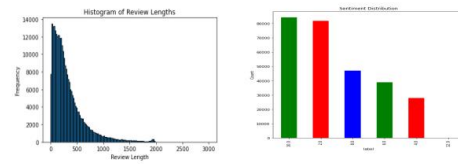
Out[222]:

	Age	Condition	Date	Drug	DrugId	EaseofUse	Effectiveness	Reviews	Satisfaction	Sex	Sides	UsefulCount
0	75 or over	Stuffy Nose	9/21/2014	25dph-7.5peh	146724	5	5	I'm a retired physician and of all the meds I ...	5	Male	Drowsiness, dizziness, dry mouth /nose/thro...	0
1	25-34	Cold Symptoms	1/13/2011	25dph-7.5peh	146724	5	5	cleared me right up even with my throat hurtin...	5	Female	Drowsiness, dizziness, dry mouth /nose/thro...	1
2	65-74	Other	7/16/2012	warfarin (bulk) 100 % powder	144731	2	3	why did my PTINR go from a normal of 2.5 to ov...	3	Female		0
3	75 or over	Other	9/23/2010	warfarin (bulk) 100 % powder	144731	2	2	FALLING AND DON'T REALISE IT	1	Female		0
4	35-44	Other	1/6/2009	warfarin (bulk) 100 % powder	144731	1	1	My grandfather was prescribed this medication ...	1	Male		1

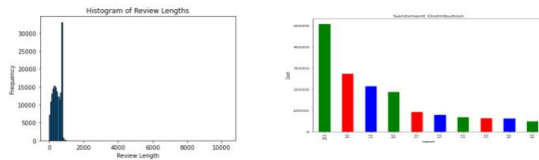
- Exploratory data analysis is performed



DrugLib review and rating distribution



Webmed review and rating distribution



DrugCom review and rating distribution

- Data cleaning -

- missing rows dropped
- special characters removed
- new lines removed
- double space removed

```

jupyter FinalProject Last Checkpoint: 22/06/2023 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted
+ < > Run Code
In [201]: #dropping missing values
dataDrugsComTrain.dropna(axis = 0, inplace = True)
dataDrugsComTrain.isna().sum()

Out[201]: Unnamed: 0      0
drugName      0
condition     0
review        0
rating        0
date          0
usefulCount   0
dtype: int64

In [202]: #white space handling
for i in dataDrugsComTrain:
    dataDrugsComTrain[i]=np.where(dataDrugsComTrain[i]==" ",np.NaN,dataDrugsComTrain[i])

dataDrugsComTrain.isna().sum()

Out[202]: Unnamed: 0      0
drugName      0
condition     0
review        0
rating        0
date          0
usefulCount   0
dtype: int64

In [203]: #removing whitespaces
dataDrugsComTrain.dropna(axis=0, inplace = True)
dataDrugsComTrain.isna().sum()
  
```



```
In [205]: # Function to remove special characters, symbols, and numbers from a string
import re
def remove_special_characters(input_string):
    return re.sub(r'[^A-Za-z\s.]', '', input_string)
```

```
In [206]: dataDrugsComTrain['review'] = dataDrugsComTrain['review'].apply(remove_special_characters)
```

```
In [207]: dataDrugsComTrain.shape
```

```
Out[207]: (160398, 7)
```

```
In [208]: dataDrugsComTrain.columns
```

```
Out[208]: Index(['Unnamed: 0', 'drugName', 'condition', 'review', 'rating', 'date',
                'usefulCount'],
                dtype='object')
```

```
In [209]: df2_selected=dataDrugsComTrain[["review", "rating"]]
```

```
In [210]: df2_selected.shape
```

```
Out[210]: (160398, 2)
```

● Combination of three datasets into one

Combining three dataset values on rating and review columns

```
In [246]: # Combine the selected columns from all three DataFrames into a new DataFrame
combined_df = pd.concat([df1_selected, df2_selected, df3_selected], ignore_index=True)
```

```
In [247]: combined_df.shape
```

```
Out[247]: (443628, 2)
```

Data Cleaning of combined dataset

```
In [296]: #missing values
combined_df.isna().sum()
```

```
Out[296]: review    0
          rating    0
          dtype: int64
```

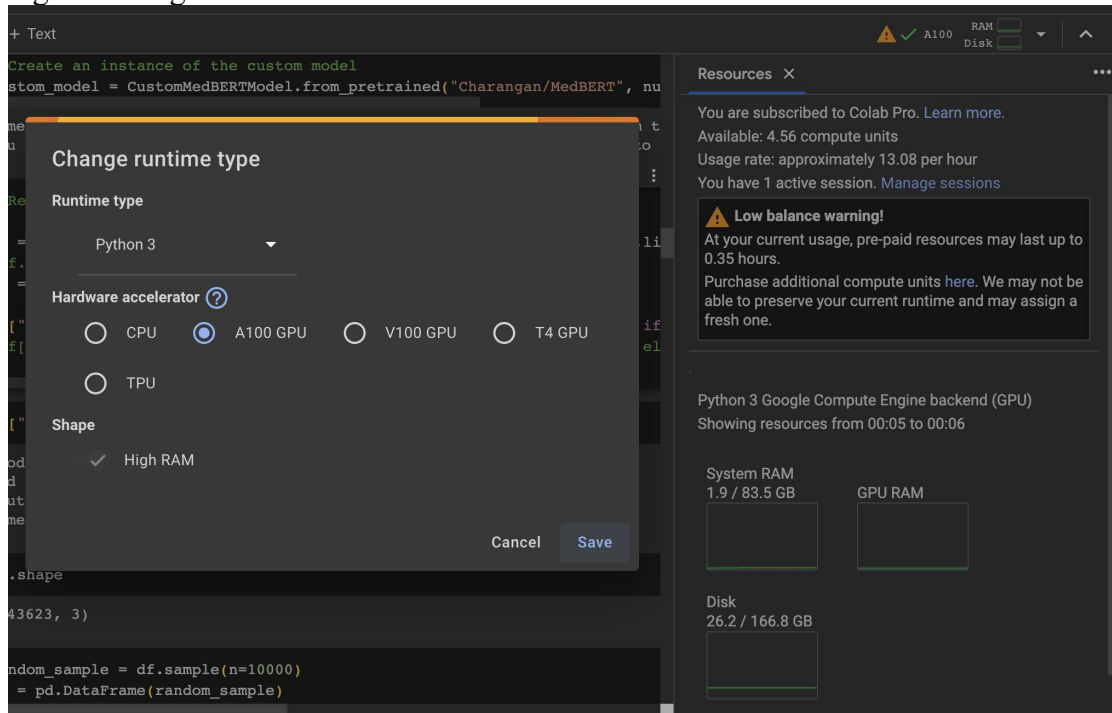
```
In [250]: #dropping missing values
combined_df.dropna(axis = 0, inplace = True)
combined_df.isna().sum()
```

```
Out[250]: review    0
          rating    0
          dtype: int64
```

● Cleaned data saved in local machine for further analysis

5.2 medBERT.de model implementation steps

- login to Google collab environment



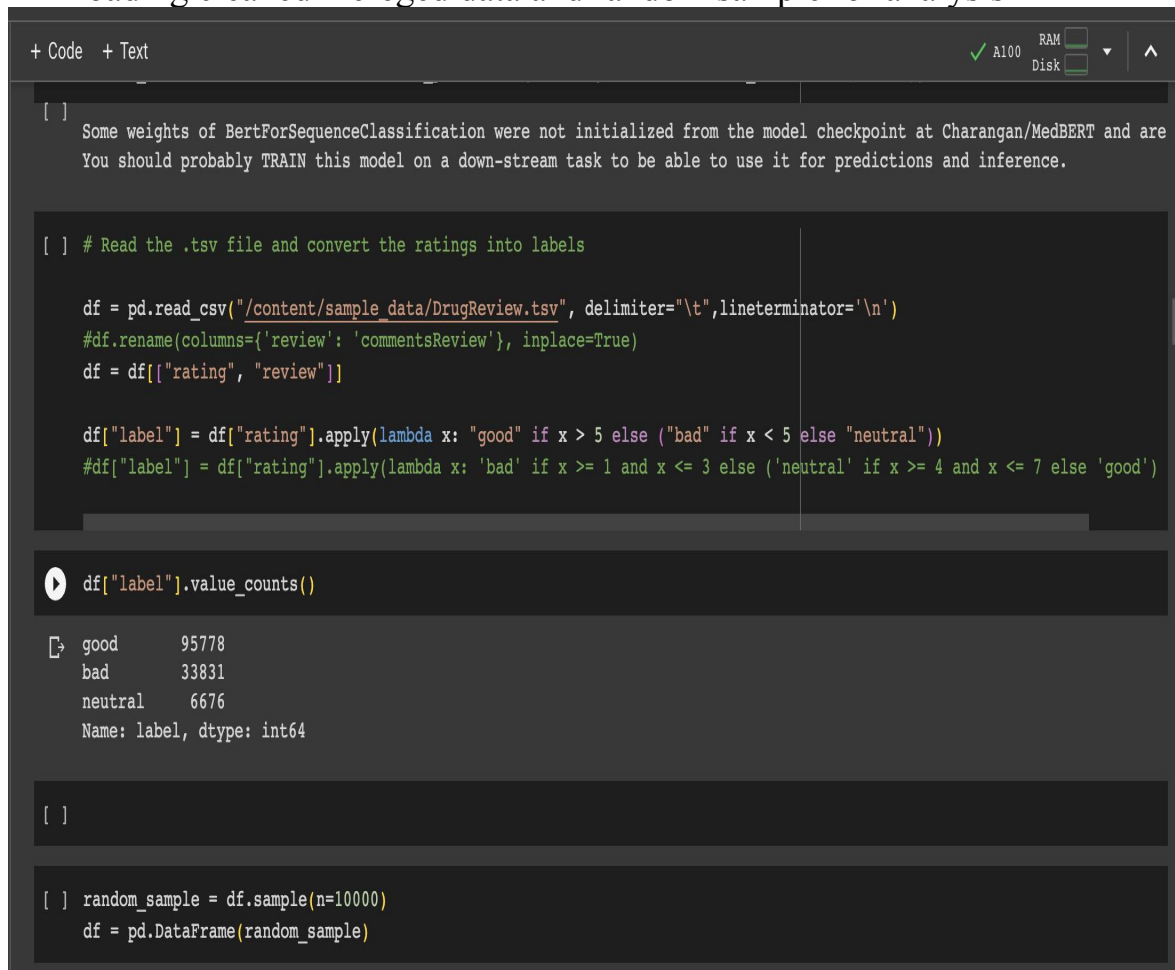
The screenshot shows the Google Colab interface. A 'Change runtime type' dialog box is open, showing the following options:

- Runtime type: Python 3
- Hardware accelerator: A100 GPU, CPU, V100 GPU, T4 GPU, TPU
- Shape: High RAM

The 'Resources' panel on the right shows the following information:

- You are subscribed to Colab Pro. Learn more.
- Available: 4.56 compute units
- Usage rate: approximately 13.08 per hour
- You have 1 active session. Manage sessions
- Low balance warning!** At your current usage, pre-paid resources may last up to 0.35 hours. Purchase additional compute units here. We may not be able to preserve your current runtime and may assign a fresh one.
- Python 3 Google Compute Engine backend (GPU)
- Showing resources from 00:05 to 00:06
- System RAM: 1.9 / 83.5 GB
- GPU RAM: [Progress bar]
- Disk: 26.2 / 166.8 GB

- Loading cleaned merged data and random sample for analysis



```
[ ] Some weights of BertForSequenceClassification were not initialized from the model checkpoint at Charangan/MedBERT and are
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.

[ ] # Read the .tsv file and convert the ratings into labels

df = pd.read_csv("/content/sample_data/DrugReview.tsv", delimiter="\t", lineterminator='\n')
#df.rename(columns={'review': 'commentsReview'}, inplace=True)
df = df[["rating", "review"]]

df["label"] = df["rating"].apply(lambda x: "good" if x > 5 else ("bad" if x < 5 else "neutral"))
#df["label"] = df["rating"].apply(lambda x: 'bad' if x >= 1 and x <= 3 else ('neutral' if x >= 4 and x <= 7 else 'good'))

df["label"].value_counts()

good    95778
bad     33831
neutral  6676
Name: label, dtype: int64

[ ]

[ ] random_sample = df.sample(n=10000)
df = pd.DataFrame(random_sample)
```

● Text preprocessing and Feature extraction

```
model with 10000 records , batch 32 epoch-10, decay weight 0.1 ☆
Help
+ Code + Text
Reconnect A100 High RAM

def tokenize(batch):
    tokenized_texts = tokenizer(batch["review"], padding='max_length', truncation=True, max_length=512)

    filtered_texts = []
    sentiment_polarities = []

    # TF-IDF feature extraction
    tfidf_vectorizer = TfidfVectorizer(max_features=5000) #
    tfidf_matrix = tfidf_vectorizer.fit_transform([tokenizer.decode(text) for text in tokenized_texts['input_ids']])

    #feature selection

    for text, tfidf_values in zip(tokenized_texts['input_ids'], tfidf_matrix):
        # Convert input_ids to tokens using tokenizer's decode method
        original_text = tokenizer.decode(text)
        # Apply lemmatization
        lemmatized_text = lemmatize_sentence(original_text)
        filtered_text = [word for word in lemmatized_text.split() if word.lower() not in stop_words]
        filtered_texts.append(' '.join(filtered_text))
        #sentiment_polarity = get_sentiment_polarity(' '.join(filtered_text))
        #sentiment_polarities.append(sentiment_polarity)

    return {
        "input_ids": tokenized_texts["input_ids"],
        "attention_mask": tokenized_texts["attention_mask"],
        "tfidf_features": tfidf_matrix.toarray(), # Convert sparse matrix to dense array
        "#sentiment_polarities": sentiment_polarities,
        "labels": [label_to_id[label] for label in batch["label"]]
    }
```

● pretrained medBERT.de Model loading from the Huggingface Hub

```
+ Code + Text
Reconnect A100 High RAM

# Define the labels and create label to id and id to label maps
labels = ["good", "bad", "neutral"]
label_to_id = {label: i for i, label in enumerate(labels)}
id_to_label = {i: label for label, i in label_to_id.items()}

# Load the tokenizer and model from the Hugging Face Hub
tokenizer = AutoTokenizer.from_pretrained("Charangan/MedBERT")
model = AutoModelForSequenceClassification.from_pretrained("Charangan/MedBERT", num_labels=len(labels))

Downloading (...)okenizer_config.json: 100% ██████████ 417/417 [00:00<00:00, 33.4kB/s]
Downloading (...)solve/main/vocab.txt: 100% ██████████ 213k/213k [00:00<00:00, 13.0MB/s]
Downloading (...)main/tokenizer.json: 100% ██████████ 669k/669k [00:00<00:00, 30.7MB/s]
Downloading (...)cial_tokens_map.json: 100% ██████████ 125/125 [00:00<00:00, 11.4kB/s]
Downloading (...)ve/main/config.json: 100% ██████████ 682/682 [00:00<00:00, 65.7kB/s]
Downloading pytorch_model.bin: 100% ██████████ 436M/436M [00:00<00:00, 539MB/s]

Some weights of BertForSequenceClassification were not initialized from the model checkpoint at Charangan/MedBERT and a
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

● Model training

```

+ Code + Text
Reconnect A100 High RAM ^

[ ] # Create a Trainer instance and train the model
trainer = Trainer(
    model=custom_model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset,
    compute_metrics=compute_metrics,
)

▶ trainer.train()

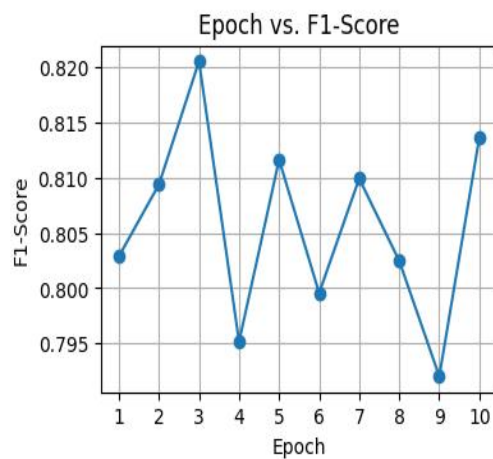
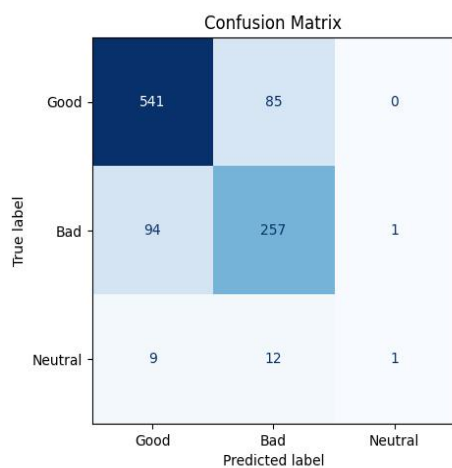
/usr/local/lib/python3.10/dist-packages/transformers/optimization.py:411: FutureWarning: This implementation of AdamW is
warnings.warn(
[1250/1250 14:54, Epoch 5/5]

Epoch Training Loss Validation Loss Accuracy Precision Recall F1 Confusion Matrix
1 No log 0.785771 0.640000 0.801428 0.640000 0.706980 [[439, 86, 5], [54, 198, 2], [113, 100, 3]]
2 0.823400 0.736825 0.696000 0.789174 0.696000 0.730998 [[465, 49, 16], [45, 192, 17], [105, 72, 39]]
3 0.823400 0.758925 0.695000 0.749428 0.695000 0.716140 [[459, 42, 29], [41, 180, 33], [96, 64, 56]]
4 0.493200 0.815804 0.694000 0.734487 0.694000 0.710476 [[457, 36, 37], [36, 175, 43], [99, 55, 62]]
5 0.493200 0.845057 0.693000 0.713303 0.693000 0.701832 [[449, 33, 48], [32, 170, 52], [86, 56, 74]]

Trainer is attempting to log a value of "[[439, 86, 5], [54, 198, 2], [113, 100, 3]]" of type <class 'list'> for key "eva
Trainer is attempting to log a value of "[[465, 49, 16], [45, 192, 17], [105, 72, 39]]" of type <class 'list'> for key "e
Trainer is attempting to log a value of "[[459, 42, 29], [41, 180, 33], [96, 64, 56]]" of type <class 'list'> for key "e

```

● Results



Accuracy : 0.812
 precision :0.8150
 recall :0.812
 f1_score : 0.83

